</

Analyzing eCommerce Business Performance with SQL

/>

MUHAMMAD IFZAL ASRIL

1 0 1 1   0 1 1   0 1   1 0 1 1 0 0 1   1 0   1 1 0 1 1   0 1 1   0 1   1 1 0 1 1 0   1 1 0 1 1 1   1 1 0 1

Description

This mini-project is part of Rakamin Academy's Data Science Bootcamp.

It analyzes the performance of an eCommerce business,
including growth, stagnation, and decline.

Customer activity on the eCommerce platform is a key metric for measuring performance.

In addition, I examined several customer activity metrics, including.
the number of active customers,
the number of new customers,
the number of repeat customers
and also the average number of transactions per customer per year.

1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

Metrics

- Average Monthly Active User
- New Customers
- Repeating Customers
- Average Orders per Customers

Average Monthly Active Users

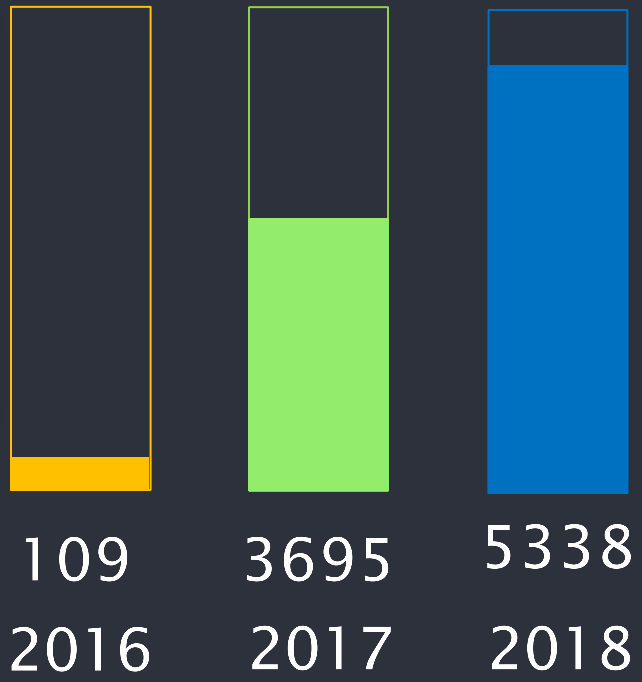
The initial step is to generate a temporary table utilizing a subquery in the FROM clause.
The purpose of this table is to count the number of monthly active users (MAU),
which refers to the unique customers who regularly place orders within a certain month.

To extract the necessary timestamp components,
specifically the year and month, the date_part function is utilized.

After obtaining the MAU number for each month, perform a larger aggregation to get the average MAU for every month. Then, to obtain the average MAU for each year, execute another larger aggregation, grouped by year.

Average Monthly Active Users

	year	average_monthly_active_user
1	2016	109
2	2017	3695
3	2018	5338



New Customers

The temporary table contains information regarding the time of each customer's initial purchase.

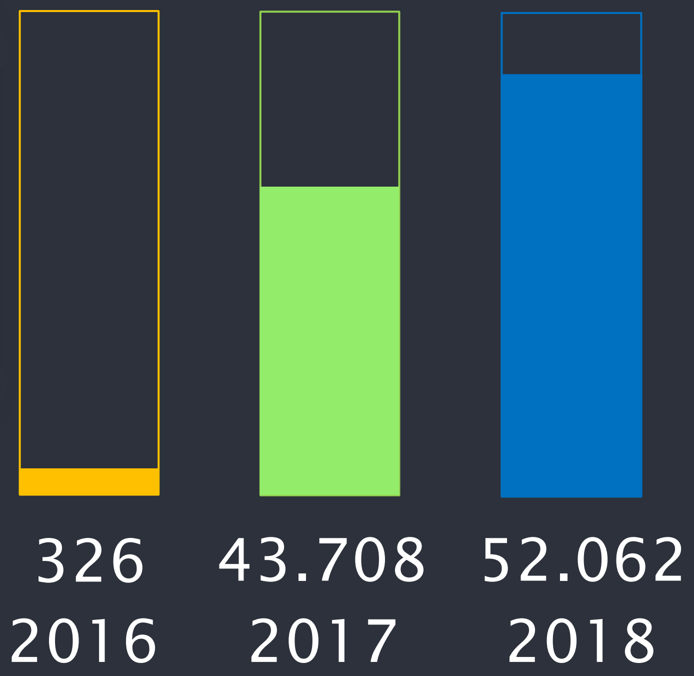
To obtain this data efficiently, use the MIN function on the order_purchase_timestamp column to determine the earliest date for each customer.

Once you have the first order date for each customer, the next step involves extracting the year and determining if there is a purchase date for each customer.

The next step involves extracting the year and tallying the number of customers for each year.
This final number indicates the amount of new customers acquired annually.

New Customers

	year	new_customers
1	2016	326
2	2017	43708
3	2018	52062



Repeating Customer

In Subtask 3, I created a temporary table that displays each customer's yearly order frequency.

I used COUNT calculations to group by year and customer_unique_id, aiming to identify customers, who place repeat orders.

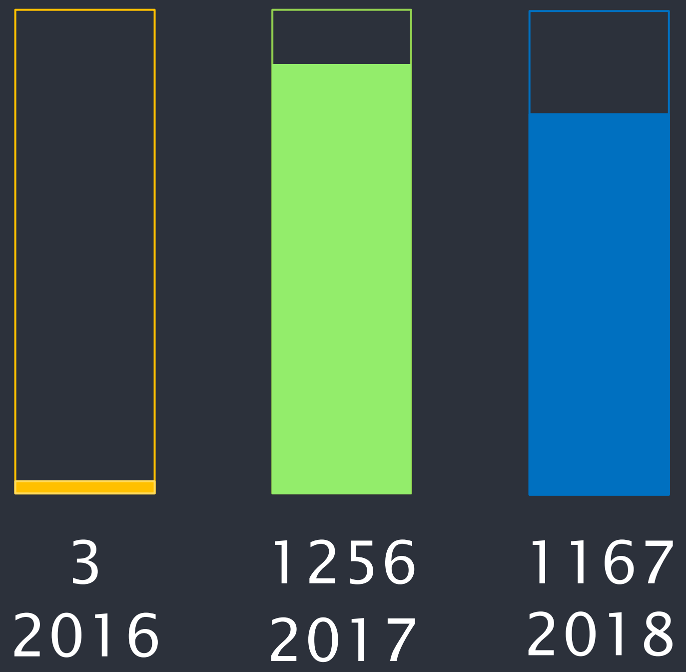
I used COUNT calculations to group by year and customer_unique_id, aiming to identify customers who place repeat orders.

Using the HAVING filter, I filtered for customers who placed more than one order.

Finally, I aggregated COUNTs grouped by year to determine how many customers made repeat orders (i.e., more than one) each year.

Repeating Customers

	year	repeating_customers
1	2016	3
2	2017	1256
3	2018	1167



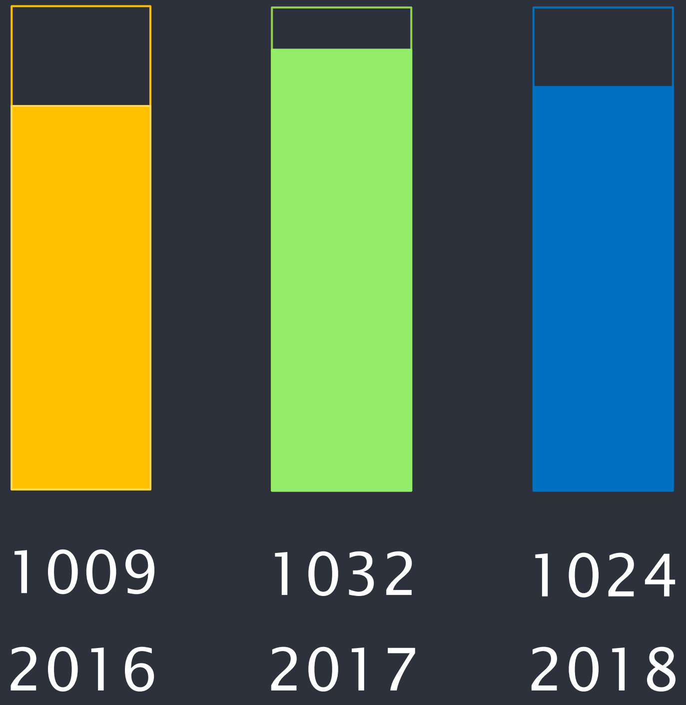
Average Orders per Customers

After creating a table with order frequency information for every customer per year,
and can perform additional aggregation by calculating the average frequency using the Frequency column.

Then, group the data by year to obtain the average number of orders for each customer in each year.
This approach allows for precise analysis and insights into customer behavior and trends.

Average Orders per Customers

	year double precision	avg_orders_per_customers numeric
1	2016	1.009
2	2017	1.032
3	2018	1.024



	year	average_mau	new_customers	repeating_customers	avg_orders_per_customers
1	2016	108.67	326	3	1.009
2	2017	3694.83	43708	1256	1.032
3	2018	5338.20	52062	1167	1.024

According to the aggregated metrics,
2016 had the lowest average number of users compared to the other years.


THANK YOU!